

# Minuit

- ▶ Minimizzazione di una funzione
- ▶ Fasi del processo
- ▶ Codifica della funzione
- ▶ Uso di Minuit

## Cos'è Minuit

Minuit è una libreria di funzioni con **interfaccia Fortran** che permette di trovare quali sono gli **argomenti** di una funzione numerica per i quali la funzione stessa dà il risultato **minimo**

Informazioni disponibili su web: <http://wwwasdoc.web.cern.ch/wwwasdoc/minuit/minmain.html>

La funzione deve essere, ovviamente, scritta dall'utente in un linguaggio interfacciabile con il Fortran, quindi non solo Fortran ma anche C e C++ ; la funzione può avere un **numero qualsiasi** di argomenti da cui calcolare un unico valore numerico. Sia gli argomenti che il risultato devono essere rappresentati da variabili in **doppia precisione**; nel processo numerico viene attribuita una corrispondente accuratezza al calcolo della funzione stessa, salvo **diversa indicazione** dell'utente.

Ai valori di argomento corrispondenti al minimo viene attribuito un **errore**, dato dalla variazione per la quale la funzione dà un risultato maggiore del minimo di una quantità prestabilita, pari all'unità se non specificata.

È possibile poi scegliere il livello di compromesso tra velocità ed accuratezza.

## Fasi del processo

Il processo di minimizzazione avviene in varie fasi:

- ▶ Definizione delle **variabili**: nome, valore iniziale, passo, eventuali limiti
- ▶ **Inizializzazione** della funzione numerica ( lettura di parametri e/o istogrammi ): la funzione viene chiamata una prima volta, con i parametri iniziali
- ▶ Ricerca del **minimo**: la funzione viene chiamata ripetutamente
- ▶ Calcolo degli **errori**: la funzione viene ancora chiamata ripetutamente
- ▶ **Finalizzazione**: la funzione viene chiamata per l'ultima volta, con i parametri corrispondenti al minimo trovato
- ▶ Estrazione dei **risultati**

Queste varie fasi vengono eseguite seguendo le **istruzioni** date dall'utente, in due possibili modi:

- ▶ “**Data driven**” mediante una serie di istruzioni date **durante l'esecuzione**
- ▶ “**Fortran callable**” mediante una serie di chiamate a **subroutine** Fortran

# Funzione FCN

La funzione numerica da minimizzare viene codificata in una subroutine, spesso chiamata “FCN”, con vari argomenti:

```
subroutine fcn(npar,grad,fval,xval,iflag,futil)
```

- ▶ npar è il **numero di variabili** d'ingresso
- ▶ grad è un array contenente le **derivate** della funzione ( opzionali )
- ▶ fval è il **risultato** della funzione
- ▶ xval è un array contenente i **parametri** con cui la funzione viene calcolata
- ▶ iflag è un **segnale** pari a
  - ↪ 1 quando la funzione viene chiamata la prima volta
  - ↪ 3 quando la funzione viene chiamata l'ultima volta
  - ↪ 4 durante la minimizzazione
- ▶ futil rappresenta una **funzione** ausiliaria ( raramente utilizzata )

Tutti gli argomenti, in ingresso ed in uscita, sono, quando non interi, in **doppia precisione**.

## Esecuzione

Alle varie fasi del processo di minimizzazione corrispondono vari **comandi** che possono venire letti durante l'esecuzione, dall'ingresso standard o da file, oppure eseguiti mediante singole chiamate a subroutine Fortran.

```
call mintio(luni,luno,lune)
call minuit(fcn,futil)
```

Le istruzioni vengono lette dall'unità specificata nella chiamata a `mintio`; la **lista dei parametri** è, ovviamente, all'inizio, anche se può essere preceduta da un **titolo**. Ogni parametro viene definito in **una linea** in cui vengono specificati:

- ~> un numero intero d'identificazione
- ~> un nome dato come stringa
- ~> il valore iniziale
- ~> il passo
- ~> eventuali minimo e massimo

# Comandi

Dopo i parametri va inserita una **linea vuota** e successivamente i vari comandi.

- ~> **set err X** per impostare il margine entro cui calcolare gli errori
- ~> **set eps 1.0e-10** per impostare la precisione del calcolo
- ~> **call lflag** per chiamare la funzione con il valore dato di IFLAG
- ~> **migrad** per eseguire la minimizzazione
- ~> **minos** per un accurato calcolo degli errori
- ~> **fix N** per fissare un parametro
- ~> **rel N** per rilasciare un parametro
- ~> **set par N X** per impostare un parametro
- ~> **scan Np Ns Xmin Xmax** per calcolare la funzione modificando il parametro Np
- ~> **mncont N1 N2 Np** per tracciare la linea corrispondente all'errore per i due parametri
- ~> ...

## Istruzioni da Fortran

I comandi possono venire eseguiti **direttamente da Fortran** mediante la chiamata alla routine **mnexcm**:

```
call mnexcm(fcn, chcomm, arglis, nargs, ierror, futil)
```

Il comando viene dato nella **stringa chcomm** e i parametri vengono dati nell'**array arglis** la cui **lunghezza** è specificata in **nargs**.

Per alcuni comandi esiste una subroutine dedicata:

```
call mninit(luni, luno, lune)
call mnseti(chtit)
call mnparm(ipar, chname, xini, step, xmin, xmax, ierr)
...
call mnpout(ipar, chname, xval, erro, xmin, xmax, ivar)
call mnerrs(ipar, errl, errh, errp, cglo)
```

## Chiamata da C++

Come tutte le funzioni Fortran anche le subroutine di Minuit possono venire **chiamate da C++** ; qualche attenzione è richiesta nella dichiarazione per la presenza di un **pointer a funzione**.

Una semplificazione può venire da **ROOT**: è possibile creare un oggetto di classe **TMinuit**, le cui funzioni corrispondono alle subroutine di Minuit. In questo caso la funzione **FCN** non viene passata come argomento ad ogni chiamata a `mnexcm` ma viene **comunicata** una sola volta **all'inizio**:

```
TMinuit rmin(npar);  
rmin.SetFCN(fcn_);  
rmin.mninit(ilun,olun,elun);  
rmin.mnparm(ipar,strn,xini,step,xmin,xmax,ierr);  
rmin.mnexcm(cstr,argv,argvn,ierr);
```